Nick Travers

228B Lecture Notes 01/29/09

Recall we are trying to solve the the heat equation $u_t = bu_{xx}$. Using the Crank-Nicolson discretization we get

$$\left(I - \frac{b\Delta t}{2}L\right)u^{n+1} = \left(I + \frac{b\Delta t}{2}L\right)u^n \tag{1}$$

In 1D this can be solved easily and directly using Gaussian elimination since we have a tridiagonal matrix. However, in higher dimensions this is NOT the case. Last time we discussed possibly using CG, SOR, or MG. All of these should work, but what sort of convergence can we expect?

We Consider Two Extreme Cases:
(1) $\Delta t \to 0$, h fixed
(2) $\Delta t \to \infty$, h fixed

case (1):
If $\Delta t \to 0$ then $I - \frac{b\Delta t}{2}L \to I$
I is, of course, very easy to invert so we expect convergence in 1 step from an iterative method.

case (2):
If $\Delta t \to \infty$ then $I - \frac{b\Delta t}{2}L$ blows up. But we are actually trying to solve:

$$\left(I - \frac{b\Delta t}{2}L\right)u^{n+1} = \left(I + \frac{b\Delta t}{2}L\right)u^n$$

$$-\frac{b\Delta t}{2}Lu^{n+1} = \frac{b\Delta t}{2}Lu^n + o(1/\Delta t)$$

$$-Lu^{n+1} = Lu^n = r$$

This is just the discrete Poisson equation. So at best our iterative methods (CG,SOR,MG) take 1 iteration, and at worst they take the same number of iterations as required to solve the Poisson equation.

Why is using $\Delta t \to \infty$ interesting? When using Crank-Nicolson and taking $\Delta t \to 0$ with $\Delta t/h$ constant, $\|L\| \to \infty$ ($L \sim 1/h^2$). Thus, $\|\Delta t L\| \to \infty$ as $h \to 0$ ($\Delta t L \sim 1/h$). So again we are essentially solving the discrete Poisson equation at each time step.

As $\Delta t \to 0$, with $\Delta t/h$ fixed, the condition number of the matrix $A = I - \frac{b\Delta t}{2} L$ increases from that of the identity I (small) to that of the discrete Laplacian L (large). Does this suggest that convergence will slow down? Not necessarily since we have a good initial guess $u^{n+1} \sim u^n$ and this approximation gets even better as $\Delta t \to 0$.

Main Point

All iterative methods (CG,SOR,MG) for solving Laplace's equation generally work even better when solving the diffusion equation because:

1. The matrix is better conditioned.
2. We have a much better initial guess $u^{n+1} \sim u^n$.

Example

Using MG to solve $\Delta u = f(x)$ or $u_t = b\Delta u$ on a 64 by 64 grid with periodic BC, $\nu_1 = \nu_2 = 1$ and GS-RB as a smoother.

(1) <u>Poisson Equation</u>: convergence factor $\rho \approx 0.16$, 7-8 iterations to reduce error by factor of 10^6.

(2) <u>Diffusion Equation</u>:
$b = 1.00$, $\rho \approx 0.110$, 6-7 iterations to reduce error by factor of 10^6.
$b = 0.10$, $\rho \approx 0.050$, 4-5 iterations to reduce error by factor of 10^6.
$b = 0.01$, $\rho \approx 0.035$, 4 iterations to reduce error by factor of 10^6.

The actual number of iterations required in practice might be even smaller because we have a good initial guess. Also, if the solution is changing slowly in time, the number of iterations is expected to be smaller.

Direct Methods, $O(M)$ work:
There are also other methods to approximately solve

$$\left(I - \frac{b\Delta t}{2} L\right) u^{n+1} = \left(I + \frac{b\Delta t}{2} L\right) u^n$$

directly (i.e. approximately invert matrix without iterating). The work required is $O(M)$, where M is the number of grid points. This is $O(N^d)$, where d is the number of spatial dimensions and $N = O(h^{-1})$. These methods exploit the fact that we are time stepping, and are thus not available for the Poisson equation. We describe two basic methods:

1. ADI Methods (Alternating Direction Implicit)
2. LOD Methods (Locally one-dimensional)

Both of these are examples of operator splitting (or fractional step) methods. We write the Laplacian operator in two dimensions as the sum of two one-dimensional Laplacians: $\Delta u = \dfrac{\partial^2 u}{\partial x^2} + \dfrac{\partial^2 u}{\partial x^2}$, and similarly we split the discrete Laplacian as $L = L_x + L_y$.

LOD Method:
Essentially we diffuse only in the x-direction first and then only in the y-direction. This gives a two step process:

$$\left(I - \frac{b\Delta t}{2} L_x\right) u^{*} = \left(I + \frac{b\Delta t}{2} L_x\right) u^n$$

$$\left(I - \frac{b\Delta t}{2} L_y\right) u^{n+1} = \left(I + \frac{b\Delta t}{2} L_y\right) u^{*}$$

ADI Method:
This is similar to the LOD method but we instead couple the x and y components in each $1/2$ time step. This has some advantages over the LOD method, as we will see later:

$$\left(I - \frac{b\Delta t}{2} L_x\right) u^{*} = \left(I + \frac{b\Delta t}{2} L_y\right) u^n$$

$$\left(I - \frac{b\Delta t}{2} L_y\right) u^{n+1} = \left(I + \frac{b\Delta t}{2} L_x\right) u^{*}$$

work estimate:
Diffusion in $x \longrightarrow$ Solve N_y tridiagonal systems
Diffusion in $y \longrightarrow$ Solve N_x tridiagonal systems
Each of N_y tridiagonal systems is size N_x
Each of N_x tridiagonal systems is size N_y
Total work is $O(N_xN_y + N_yN_x) = O(M)$

Stability:
We perform von Neumann Analysis for the LOD Method. In two spatial dimensions we Fourier transform both the x and y variables, resulting in two wave numbers, ξ_1 and ξ_2. To simplify the notation, we let $mu = b\Delta t/h^2$.

$$\hat{u^*} = \left(\frac{1 - 4\mu \sin^2{(\xi_1 h/2)}}{1 + 4\mu \sin^2{(\xi_1 h/2)}}\right)\hat{u^n}$$

$$\hat{u^{n+1}} = \left(\frac{1 - 4\mu \sin^2{(\xi_2 h/2)}}{1 + 4\mu \sin^2{(\xi_2 h/2)}}\right)\hat{u^*}$$

combining these two equations gives:

$$\hat{u^{n+1}} = \left(\frac{(1 - 4\mu \sin^2{(\xi_1 h/2)})(1 - 4\mu \sin^2{(\xi_2 h/2)})}{(1 + 4\mu \sin^2{(\xi_1 h/2)})(1 + 4\mu \sin^2{(\xi_2 h/2)})}\right)\hat{u^n} = \rho(\xi_1, \xi_2)\hat{u^n}.$$

Clearly, $\rho(\xi_1, \xi_2) \leq 1$. Hence, the method is (unconditionally) stable. The result $\hat{u^{n+1}} = \rho(\xi_1, \xi_2)\hat{u^n}$ is exactly the same for the ADI Method, so ADI is stable as well.

Consistency:
Showing the consistency of LOD and ADI is actually harder than showing stability. We will show consistency only for ADI. To use the same proof for LOD we must assume L_x and L_y commute. This is true for periodic BC and constant coefficients but NOT in general. We will return to LOD after considering fractional step methods more generally.

To show consistency for ADI we apply $(I + \frac{b\Delta t}{2}L_x)$ to both sides of the equation for the first half time step in ADI and use that $(I - \frac{b\Delta t}{2}L_x)$ and $(I + \frac{b\Delta t}{2}L_x)$ commute. Again to simplify notation, we define $\beta = b\Delta t/2$.

$$(I - \beta L_x)u^* = (I + \beta L_y)u^n$$

$$
\begin{aligned}
(I + \beta L_x)(I - \beta L_x)u^* &= (I + \beta L_x)(I + \beta L_y)u^n \\
(I - \beta L_x)(I + \beta L_x)u^* &= (I + \beta L_x)(I + \beta L_y)u^n \\
(I - \beta L_x)(I - \beta L_y)u^{n+1} &= (I + \beta L_x)(I + \beta L_y)u^n \\
(I - \beta L_x - \beta L_y + \beta^2 L_x L_y)u^{n+1} &= (I + \beta L_x + \beta L_y + \beta^2 L_x L_y)u^n \\
(I - \beta L u^{n+1}) &= (I + \beta L u^n)u^n - \beta^2 L_x L_y (u^{n+1} - u^n)
\end{aligned}
$$

This looks like the Crank-Nicolson discretization with an additional term on the right side of the equation. We now rearrange this equation to look more like the discretized PDE:

$$\frac{u^{n+1} - u^n}{\Delta t} = \frac{b}{2}\Big(Lu^{n+1} + Lu^n\Big) - \frac{b^2 \Delta t}{4} L_x L_y \big(u^{n+1} - u^n\big)$$

By performing a Taylor expansion, we see that this extra term introduces an $\mathcal{O}(\Delta t^2)$ error. Overall the scheme is still 2nd order accurate in space and time.

Note: We have shown ADI is stable and consistent, and so by the Lax-Equivalence theorem it is also convergent.